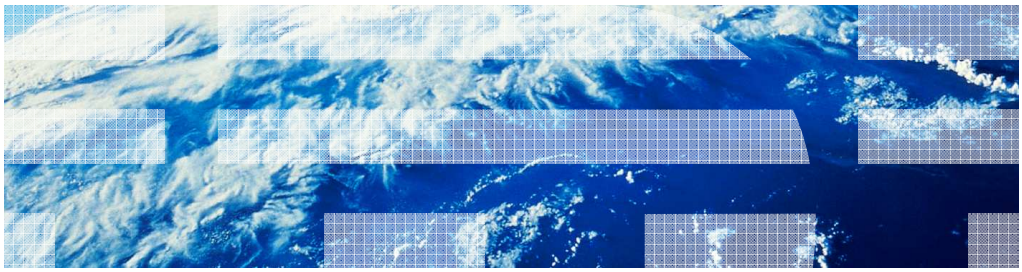
z/OS V1R13

BCP runtime diagnostics: Z/OS R13 support



Session objectives

- The objective of this session is to show how Runtime Diagnostics is configured and how it benefits a system programmer.
  - Runtime Diagnostics allows a system programmer to quickly analyze a system experiencing “sick, but not dead” symptoms.
    - Runtime Diagnostics was introduced in R12 and is now being enhanced for R13.
  - There are minimal migration issues.
  - Runtime Diagnostics should be started at IPL via an entry in COMMNDxx or by the installation's automation.
  - Modify (F) command is used to tell Runtime Diagnostics to do a run.
  - Little configuration is required.
  - The publication is listed at the end of this presentation.

Overview (1 of 3)

- Problem Statement / Need Addressed
  - Analyze a “sick, but not dead” system in a timely manner.
- Solution:
  - Runtime Diagnostics attempts to analyze a “sick, but not dead” system in a minute or so and provides suggested next steps to take.
  - Looks for evidence of “soft failures”
    - Component issues
    - Global resource contention
    - Important address space execution issues
- Benefit / Value
  - Use Runtime Diagnostics
    - When the help desk or operations reports a problem.
    - To get ready for a “bridge call”
  - Reduces the skill level needed by a system programmer when examining z/OS® for “unknown” problems.

Overview (2 of 3)

- Using Runtime Diagnostics the system programmer can quickly analyze a sick system for the following classes of problems:
  - Component problems emitted as critical messages in OPERLOG (needles in the haystack) (R12)
  - ENQ contention for system address spaces (R12)
  - Address spaces with a high local lock suspension rate (R12)
  - Address spaces using high CPU (R12)
  - Address spaces that appear to be in a TCB enabled loop (R12)
  - GRS latch Contention (R13)
  - z/OS File System latch Contention (R13)
- Runtime Diagnostics recommends next actions the system programmer should take
  - Potentially, what jobs might need to be cancelled
  - Further investigation on class of resources or a single address space using a monitor like RMF™ or Tivoli® Omegamon

Overview (3 of 3)

- In R13, Runtime Diagnostics is invoked via a MODIFY command from the console
  - F HZR,ANALYZE
- The output of Runtime Diagnostics is a multi-line WTO to the operator console
  - The Runtime Diagnostics response is issued to the console that issued the command
    - If the MCS console that issued the command has an out-of-line display area setup (via a K A,xx) the output will be displayed in the display area
  - The output of Runtime Diagnostics can also be directed to a sequential dataset
- Examples for Runtime Diagnostics are provided later in this presentation

Examples (1 of 2)

- Success

```
f hzr,analyze
HZR0200I RUNTIME DIAGNOSTICS RESULT 974
SUMMARY: SUCCESS
 REQ: 001 TARGET SYSTEM: SY1      HOME: SY1      2010/12/21 - 11:30:57
 INTERVAL:  60 MINUTES
 EVENTS:
  FOUND: 05 - PRIORITIES: HIGH:05  MED:00  LOW:00
  TYPES: CF:04
  TYPES: HIGHCPU:01
```

- Status message showing HZR pointing to a remote system

```
f hzr,analyze
HZR0200I RUNTIME DIAGNOSTICS RESULT 593
SUMMARY: SUCCESS - NO EVENTS FOUND
 REQ: 001 TARGET SYSTEM: SYS3      HOME: SY1      2011/08/05 -
09:15:35
 INTERVAL:  60 MINUTES
 EVENTS:
  FOUND: 00 - PRIORITIES: HIGH:00  MED:00  LOW:
  PROCESSING BYPASSED:
   OMVS.......SPECIFIED TARGET SYSTEM IS NOT THE HOME SYSTEM.
   LATCHES....SPECIFIED TARGET SYSTEM IS NOT THE HOME SYSTEM.
   LOOP.......SPECIFIED TARGET SYSTEM IS NOT THE HOME SYSTEM.
   HIGHCPU....SPECIFIED TARGET SYSTEM IS NOT THE HOME SYSTEM.
   LOCK.......SPECIFIED TARGET SYSTEM IS NOT THE HOME SYSTEM.
```

Examples (2 of 2)

- Qualified Success – Example of Operlog not connecting in R12

```
f hzr,analyze
HZR0200I RUNTIME DIAGNOSTICS RESULT 751
SUMMARY: QUALIFIED SUCCESS - SOME PROCESSING FAILED
 REQ: 001 TARGET SYSTEM: SY1      HOME: SY1      2010/12/21 - 11:25:55
 INTERVAL:  60 MINUTES
 EVENTS:
  FOUND: 02 - PRIORITIES: HIGH:02  MED:00  LOW:00
  TYPES: HIGHCPU:01
  TYPES: LOOP:01
  PROCESSING FAILURES:
   OPERLOG....IXGCONN REQ=CONNECT ERROR.......RC=00000008 RS=0000080B
```

- Success – Example of Operlog not connecting in R13 (logic changed in R13 to issue SUCCESS for this scenario)

```
f hzr,analyze
HZR0200I RUNTIME DIAGNOSTICS RESULT 239
SUMMARY: SUCCESS
 REQ: 001 TARGET SYSTEM: SY1      HOME: SY1      2011/08/05 - 09:15:19
 INTERVAL:  60 MINUTES
 EVENTS:
  FOUND: 01 - PRIORITIES: HIGH:01  MED:00  LOW:00
  TYPES: HIGHCPU:01
 PROCESSING BYPASSED:
  OPERLOG....OPERLOG IS NOT ACTIVE.
```

Example – Critical component message analysis (R12)

- Component-specific, critical messages in OPERLOG
  - "Needles in a haystack"
  - Looks one hour back, if available
- For some messages, additional analysis done
  - Groups related messages into a single event
  - Weeds out shortage and relieved critical messages
  - In some cases, will only show last message if a critical message for the same resource name is repeated, say every 10 minutes
- Message summary found listed in Runtime Diagnostics output

```
EVENT 02: HIGH - CF           - SYSTEM: SY1      2011/02/15 - 14:47:03
IXC585E STRUCTURE LIST01 IN COUPLING FACILITY TESTCFN,
PHYSICAL STRUCTURE VERSION C7565A8D E48F6410,
IS AT OR ABOVE STRUCTURE FULL MONITORING THRESHOLD OF  80%.
ENTRIES:  IN-USE:         491 TOTAL:         583,  84% FULL
ELEMENTS: IN-USE:         508 TOTAL:        1167,  43% FULL
  ERROR: INDICATED STRUCTURE IS APPROACHING FULL MONITORING THRESHOLD.
 ACTION: D XCF,STR,STRNAME=strname TO GET STRUCTURE INFORMATION.
 ACTION: INCREASE STRUCTURE SIZE OR TAKE ACTION AGAINST APPLICATION.
```

Example – Enqueue contention checking (R12)

- Looks for a system address space that is an ENQ “waiter” for over 5 seconds
- Lists both waiter and blocker
- Equivalent to D GRS,AN,WAITER

```
f hzr,analyze
HZR0200I RUNTIME DIAGNOSTICS RESULT 581
SUMMARY: SUCCESS
 REQ: 004 TARGET SYSTEM: SY1      HOME: SY1      2010/12/21 - 13:51:32
 INTERVAL:  60 MINUTES
 EVENTS:
  FOUND: 04 - PRIORITIES: HIGH:04  MED:00  LOW:00
  TYPES: HIGHCPU:01
  TYPES: LOOP:01 ENQ:01 LOCK:01
------------------------------------------------------------------------
EVENT 01: HIGH - ENQ          - SYSTEM: SY1      2010/12/21 - 13:51:32
ENQ WAITER  - ASID:0038 - JOBNAME:IBMUSER2 - SYSTEM:SY1
ENQ BLOCKER - ASID:002F - JOBNAME:IBMUSER1 - SYSTEM:SY1
QNAME: TESTENQ
RNAME: TESTOFAVERYVERYVERYVERYLOOOOOOOOOOOOOOOOOOOOONGRNAME1234567...
  ERROR: ADDRESS SPACES MIGHT BE IN ENQ CONTENTION.
 ACTION: USE YOUR SOFTWARE MONITORS TO INVESTIGATE BLOCKING JOBS AND
 ACTION: ASIDS.
```

Example – Local lock suspension (R12)

- Lists any address space where its local lock suspension time is over 50%

```
f hzr,analyze
HZR0200I RUNTIME DIAGNOSTICS RESULT 581
SUMMARY: SUCCESS
 REQ: 004 TARGET SYSTEM: SY1      HOME: SY1      2010/12/21 - 13:51:32
 INTERVAL:  60 MINUTES
 EVENTS:
  FOUND: 04 - PRIORITIES: HIGH:04  MED:00  LOW:00
  TYPES: HIGHCPU:01
  TYPES: LOOP:01 ENQ:01 LOCK:01
-------------------------------------------------------------------------
EVENT 04: HIGH - LOCK          - SYSTEM: SY1      2010/12/21 - 13:51:33
HIGH LOCAL LOCK SUSPENSION RATE - ASID:000A JOBNAME:WLM
STEPNAME:WLM      PROCSTEP:IEFPROC  JOBID:++++++++ USERID:++++++++
JOBSTART:2010/12/21 - 11:15:08
  ERROR: ADDRESS SPACE HAS HIGH LOCAL LOCK SUSPENSION RATE.
 ACTION: USE YOUR SOFTWARE MONITORS TO INVESTIGATE THE ASID.
 -----------------------------------------------------------------------
```

Example – CPU analysis (R12)

- Takes two quick samples over 1 second interval
- Any task using > 90% (R13, but 95% in R12) of a single CPU is considered a potential problem
- The usage reported might be > 100% if an address space has multiple TCBs and several are using a high percentage of the capacity of a CPU

```
f hzr,analyze
HZR0200I RUNTIME DIAGNOSTICS RESULT 581
SUMMARY: SUCCESS
 REQ: 004 TARGET SYSTEM: SY1      HOME: SY1      2010/12/21 - 13:51:32
 INTERVAL:  60 MINUTES
 EVENTS:
  FOUND: 04 - PRIORITIES: HIGH:04  MED:00  LOW:00
  TYPES: HIGHCPU:01
  TYPES: LOOP:01 ENQ:01 LOCK:01
------------------------------------------------------------------
EVENT 02: HIGH - HIGHCPU      - SYSTEM: SY1      2010/12/21 - 13:51:33
ASID CPU RATE:99%     ASID:002E   JOBNAME:IBMUSERX
STEPNAME:STEP1    PROCSTEP:        JOBID:JOB00045 USERID:IBMUSER
JOBSTART:2010/12/21 - 11:22:51
  ERROR: ADDRESS SPACE USING EXCESSIVE CPU TIME. IT MIGHT BE LOOPING.
 ACTION: USE YOUR SOFTWARE MONITORS TO INVESTIGATE THE ASID.
```

Example – Loop detection (R12)

- Investigates all tasks in all address spaces looking for TCB loops
  - Takes a snapshot of the system trace
  - Looks for consistent, repetitive activity that typically indicates a loop
- When both HIGHCPU and LOOP events occur for the same job, there is a high probability that the task in the job is in a loop.
- Normal, corrective action is to cancel the job.

```
f hzr,analyze
HZR0200I RUNTIME DIAGNOSTICS RESULT 581
SUMMARY: SUCCESS
 REQ: 004 TARGET SYSTEM: SY1      HOME: SY1      2010/12/21 - 13:51:32
 INTERVAL:  60 MINUTES
 EVENTS:
  FOUND: 04 - PRIORITIES: HIGH:04  MED:00  LOW:00
  TYPES: HIGHCPU:01
  TYPES: LOOP:01 ENQ:01 LOCK:01
---------------------------------------------------------------------
EVENT 02: HIGH - HIGHCPU       - SYSTEM: SY1      2010/12/21 - 13:51:33
ASID CPU RATE:99%     ASID:002E   JOBNAME:IBMUSERX
STEPNAME:STEP1    PROCSTEP:         JOBID:JOB00045 USERID:IBMUSER
JOBSTART:2010/12/21 - 11:22:51
  ERROR: ADDRESS SPACE USING EXCESSIVE CPU TIME. IT MIGHT BE LOOPING.
 ACTION: USE YOUR SOFTWARE MONITORS TO INVESTIGATE THE ASID.
---------------------------------------------------------------------
EVENT 03: HIGH - LOOP          - SYSTEM: SY1      2010/12/21 - 13:51:14
ASID:002E   JOBNAME:IBMUSERX   TCB:004FF1C0
STEPNAME:STEP1    PROCSTEP:         JOBID:JOB00045 USERID:IBMUSER
JOBSTART:2010/12/21 - 11:22:51
  ERROR: ADDRESS SPACE MIGHT BE IN A LOOP.
 ACTION: USE YOUR SOFTWARE MONITORS TO INVESTIGATE THE ASID.
```

Example – z/OS UNIX latch contention (R13)

- New in z/OS 1.13
- If z/OS UNIX® latch contention or waiting threads exist for > 5 minutes in z/OS UNIX, a Runtime Diagnostics OMVS event is created.
- Normal action is to issue D OMVS,W,A to get the ASID and job names of the waiters

```
F HZR,ANALYZE
HZR0200I RUNTIME DIAGNOSTICS RESULT 692
SUMMARY: SUCCESS
 REQ: 009 TARGET SYSTEM: SY1      HOME: SY1      2010/12/21 - 14:24:29
 INTERVAL:  60 MINUTES
 EVENTS:
  FOUND: 02 - PRIORITIES: HIGH:02  MED:00  LOW:00
  TYPES: OMVS:01
  TYPES: LOCK:01
-------------------------------------------------------------------------
EVENT 01: HIGH - OMVS          - SYSTEM: SY1      2010/12/21 - 14:24:29
ASID:000E - JOBNAME:OMVS
MOUNT LATCH WAITERS: 1
FILE SYSTEM LATCH WAITERS: 0
XSYS AND OTHER THREADS WAITING FOR z/OS UNIX: 1
  ERROR: z/OS UNIX MIGHT HAVE FILE SYSTEM LATCH CONTENTION.
 ACTION: D OMVS,W,A TO INVESTIGATE z/OS UNIX FILE SYSTEM LATCH
 ACTION: CONTENTION, ACTIVITY AND WAITING THREADS. USE YOUR SOFTWARE
 ACTION: MONITORS TO INVESTIGATE BLOCKING JOBS AND ASIDS.
```

Example – GRS latch contention (R13)

- New in z/OS 1.13
- Obtains latch contention information from GRS
- Omits z/OS UNIX file system latch contention
- Returns the longest waiter for each latch set

```
F HZR,ANALYZE
HZR0200I RUNTIME DIAGNOSTICS RESULT 692
SUMMARY: SUCCESS
 REQ: 002 TARGET SYSTEM: SY1      HOME: SY1      2010/12/21 - 14:32:01
 INTERVAL:  60 MINUTES
 EVENTS:
  FOUND: 02 - PRIORITIES: HIGH:02  MED:00  LOW:00
  TYPES: LATCH:02
---------------------------------------------------------------------
EVENT 01: HIGH - LATCH        - SYSTEM: SY1      2010/12/21 - 14:32:01
LATCH SET NAME: SYSTEST.LATCH_TESTSET
LATCH NUMBER:3          CASID:0039  CJOBNAME:TSTLATCH
TOP WAITER - ASID:0039 - JOBNAME:TSTLATCH - TCB/WEB:004E2A70
TOP BLOCKER- ASID:0039 - JOBNAME:TSTLATCH - TCB/WEB:004FF028
  ERROR: ADDRESS SPACES MIGHT BE IN LATCH CONTENTION.
  ACTION: D GRS,AN,LATCH,DEP,CASID=0039,LAT=(SYSTEST.L*,3),DET
  ACTION: TO ANALYZE THE LATCH DEPENDENCIES. USE YOUR SOFTWARE
  ACTION: MONITORS TO INVESTIGATE BLOCKING JOBS AND ASIDS.
```

Usage and invocation

- Runtime Diagnostics should be invoked by the system programmer when unidentified problems are being experienced.
- To invoke Runtime Diagnostics, issue a “F HZR,ANALYZE” command from an operator's console.
  - Starting with z/OS R13, the Runtime Diagnostics address space must be started.
  - See “Migration and Coexistence Considerations” later in this presentation

Interactions and dependencies

- Software dependencies
  - Operlog is suggested for complete analysis.
- Hardware dependencies
  - None
- Exploiters
  - PFA will invoke runtime diagnostics to detect when a metric is too low indicating a hung address space or hung system

Migration and coexistence considerations

- Starting with z/OS R13, Runtime Diagnostics is its own address space
  - Use the new HZR proc shipped with z/OS R13.
    - Migration Action: If you used Runtime Diagnostics in z/OS 1.12, ensure you update the hzrproc to point to PGM=HZRINIT instead of PGM=HZRMAIN.
  - START HZR,SUB=MSTR
    - Invokes HZR PROC
  - Options previously on START command (R12) are now on MODIFY command (R13)
    - For example, SYSNAME option targets system other than HOME
      - Only Operlog and ENQ analysis are done for specified system
      - Example: OPTIONS=(SYSNAME=SYS2) is now on MODIFY command
- Can override HZROUT to specify a data set, for example:
  - In the HZR PROC, specify
    - //HZROUT DD DISP=SHR,DSN=MY.DATA,LRECL=121,BLKSIZE=0,RECFM=FB
  - Runtime Diagnostics uses a QSAM option to always extend (append new data to the end of the dataset) the dataset regardless of the DISP=..
  - GDGs are not recommended.
- COMMNDxx should be updated to start HZR during the IPL process.
  - An alternative is to use the installation's automation to start HZR.

Installation

- None other than those listed in MIGRATION & COEXISTENCE.

Session summary

- Runtime Diagnostics provides a quick tool for the system programmer to analyze a “sick, but not dead” system for likely symptoms to investigate.
- Configuring HZR is very easy.

Appendix - References

- z/OS V1R13 Problem Management, G325-2564-08.

Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, RMF, Tivoli, and z/OS are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.
Other company, product, or service names may be trademarks or service marks of others.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.